

Docket No. RSW920010142US1

EXPRESS MAIL LABEL NO.: <u>ET402935174US</u>	DATE OF DEPOSIT: <u>September 21, 2001</u>
<p>I hereby certify that this paper and fee are being deposited with the United States Postal Service Express Mail Post Office to Addressee service under 37 CFR §1.10 on the date indicated above and is addressed to the Assistant Commissioner of Patents, Washington D.C. 20231.</p>	
<u>Dianne Lane</u>	
NAME OF PERSON MAILING PAPER AND FEE	SIGNATURE OF PERSON MAILING PAPER AND FEE

INVENTORS: Rajesh Agarwalla, Thirumale Niranjan, Srikanth Ramamurthy, Sumanthkumar Sukumar, Yi Zhou

METHOD AND APPARATUS FOR CACHING SUBSCRIBED AND  
NON-SUBSCRIBED CONTENT IN A NETWORK DATA PROCESSING  
SYSTEM

CROSS REFERENCE TO RELATED APPLICATIONS

5 The present invention is related to an application  
entitled Method and Apparatus for Minimizing Inconsistency  
Between Data Sources in a Web Content Distribution System,  
serial no. \_\_\_\_\_, attorney docket no. RSW920010141US1,  
filed even date hereof, assigned to the same assignee, and  
10 incorporated herein by reference.

Docket No. RSW920010142US1

**1. Field of the Invention:**

The present invention relates generally to an improved data processing system, in particular to a method and apparatus for processing data. Still more particularly, the present invention provides a method, apparatus, and computer implemented instructions for caching subscribed and non-subscribed web content in a network data processing system.

**2. Background of the Invention:**

10 The Internet, also referred to as an "internetwork", is a set of computer networks, possibly dissimilar, joined together by means of gateways that handle data transfer and the conversion of messages from a protocol of the sending network to a protocol used by the receiving network. When 15 capitalized, the term "Internet" refers to the collection of networks and gateways that use the TCP/IP suite of protocols.

The Internet has become a cultural fixture as a source of both information and entertainment. Many businesses are 20 creating Internet sites as an integral part of their marketing efforts, informing consumers of the products or services offered by the business or providing other information seeking to engender brand loyalty. Many federal, state, and local government agencies are also

DO NOT PUBLISH - DRAFT

Docket No. RSW920010142US1

employing Internet sites for informational purposes, particularly agencies which must interact with virtually all segments of society such as the Internal Revenue Service and secretaries of state. Providing informational 5 guides and/or searchable databases of online public records may reduce operating costs. Further, the Internet is becoming increasingly popular as a medium for commercial transactions.

Currently, the most commonly employed method of 10 transferring data over the Internet is to employ the World Wide Web environment, also called simply "the Web". Other Internet resources exist for transferring information, such as File Transfer Protocol (FTP) and Gopher, but have not achieved the popularity of the Web. In the Web 15 environment, servers and clients effect data transaction using the Hypertext Transfer Protocol (HTTP), a known protocol for handling the transfer of various data files (e.g., text, still graphic images, audio, motion video, etc.). The information in various data files is formatted 20 for presentation to a user by a standard page description language, the Hypertext Markup Language (HTML). In addition to basic presentation formatting, HTML allows developers to specify "links" to other Web resources identified by a Uniform Resource Locator (URL). A URL is a 25 special syntax identifier defining a communications path to specific information. Each logical block of information accessible to a client, called a "page" or a "Web page", is identified by a URL. The URL provides a universal,

8  
REGISTRATION  
TESTIMONY  
EXHIBITS  
APPENDIX  
EX-3  
EX-4  
EX-5  
EX-6  
EX-7  
EX-8  
EX-9  
EX-10  
EX-11  
EX-12  
EX-13  
EX-14  
EX-15  
EX-16  
EX-17  
EX-18  
EX-19  
EX-20  
EX-21  
EX-22  
EX-23  
EX-24  
EX-25  
EX-26  
EX-27  
EX-28  
EX-29  
EX-30  
EX-31  
EX-32  
EX-33  
EX-34  
EX-35  
EX-36  
EX-37  
EX-38  
EX-39  
EX-40  
EX-41  
EX-42  
EX-43  
EX-44  
EX-45  
EX-46  
EX-47  
EX-48  
EX-49  
EX-50  
EX-51  
EX-52  
EX-53  
EX-54  
EX-55  
EX-56  
EX-57  
EX-58  
EX-59  
EX-60  
EX-61  
EX-62  
EX-63  
EX-64  
EX-65  
EX-66  
EX-67  
EX-68  
EX-69  
EX-70  
EX-71  
EX-72  
EX-73  
EX-74  
EX-75  
EX-76  
EX-77  
EX-78  
EX-79  
EX-80  
EX-81  
EX-82  
EX-83  
EX-84  
EX-85  
EX-86  
EX-87  
EX-88  
EX-89  
EX-90  
EX-91  
EX-92  
EX-93  
EX-94  
EX-95  
EX-96  
EX-97  
EX-98  
EX-99  
EX-100  
EX-101  
EX-102  
EX-103  
EX-104  
EX-105  
EX-106  
EX-107  
EX-108  
EX-109  
EX-110  
EX-111  
EX-112  
EX-113  
EX-114  
EX-115  
EX-116  
EX-117  
EX-118  
EX-119  
EX-120  
EX-121  
EX-122  
EX-123  
EX-124  
EX-125  
EX-126  
EX-127  
EX-128  
EX-129  
EX-130  
EX-131  
EX-132  
EX-133  
EX-134  
EX-135  
EX-136  
EX-137  
EX-138  
EX-139  
EX-140  
EX-141  
EX-142  
EX-143  
EX-144  
EX-145  
EX-146  
EX-147  
EX-148  
EX-149  
EX-150  
EX-151  
EX-152  
EX-153  
EX-154  
EX-155  
EX-156  
EX-157  
EX-158  
EX-159  
EX-160  
EX-161  
EX-162  
EX-163  
EX-164  
EX-165  
EX-166  
EX-167  
EX-168  
EX-169  
EX-170  
EX-171  
EX-172  
EX-173  
EX-174  
EX-175  
EX-176  
EX-177  
EX-178  
EX-179  
EX-180  
EX-181  
EX-182  
EX-183  
EX-184  
EX-185  
EX-186  
EX-187  
EX-188  
EX-189  
EX-190  
EX-191  
EX-192  
EX-193  
EX-194  
EX-195  
EX-196  
EX-197  
EX-198  
EX-199  
EX-200  
EX-201  
EX-202  
EX-203  
EX-204  
EX-205  
EX-206  
EX-207  
EX-208  
EX-209  
EX-210  
EX-211  
EX-212  
EX-213  
EX-214  
EX-215  
EX-216  
EX-217  
EX-218  
EX-219  
EX-220  
EX-221  
EX-222  
EX-223  
EX-224  
EX-225  
EX-226  
EX-227  
EX-228  
EX-229  
EX-230  
EX-231  
EX-232  
EX-233  
EX-234  
EX-235  
EX-236  
EX-237  
EX-238  
EX-239  
EX-240  
EX-241  
EX-242  
EX-243  
EX-244  
EX-245  
EX-246  
EX-247  
EX-248  
EX-249  
EX-250  
EX-251  
EX-252  
EX-253  
EX-254  
EX-255  
EX-256  
EX-257  
EX-258  
EX-259  
EX-260  
EX-261  
EX-262  
EX-263  
EX-264  
EX-265  
EX-266  
EX-267  
EX-268  
EX-269  
EX-270  
EX-271  
EX-272  
EX-273  
EX-274  
EX-275  
EX-276  
EX-277  
EX-278  
EX-279  
EX-280  
EX-281  
EX-282  
EX-283  
EX-284  
EX-285  
EX-286  
EX-287  
EX-288  
EX-289  
EX-290  
EX-291  
EX-292  
EX-293  
EX-294  
EX-295  
EX-296  
EX-297  
EX-298  
EX-299  
EX-300  
EX-301  
EX-302  
EX-303  
EX-304  
EX-305  
EX-306  
EX-307  
EX-308  
EX-309  
EX-310  
EX-311  
EX-312  
EX-313  
EX-314  
EX-315  
EX-316  
EX-317  
EX-318  
EX-319  
EX-320  
EX-321  
EX-322  
EX-323  
EX-324  
EX-325  
EX-326  
EX-327  
EX-328  
EX-329  
EX-330  
EX-331  
EX-332  
EX-333  
EX-334  
EX-335  
EX-336  
EX-337  
EX-338  
EX-339  
EX-340  
EX-341  
EX-342  
EX-343  
EX-344  
EX-345  
EX-346  
EX-347  
EX-348  
EX-349  
EX-350  
EX-351  
EX-352  
EX-353  
EX-354  
EX-355  
EX-356  
EX-357  
EX-358  
EX-359  
EX-360  
EX-361  
EX-362  
EX-363  
EX-364  
EX-365  
EX-366  
EX-367  
EX-368  
EX-369  
EX-370  
EX-371  
EX-372  
EX-373  
EX-374  
EX-375  
EX-376  
EX-377  
EX-378  
EX-379  
EX-380  
EX-381  
EX-382  
EX-383  
EX-384  
EX-385  
EX-386  
EX-387  
EX-388  
EX-389  
EX-390  
EX-391  
EX-392  
EX-393  
EX-394  
EX-395  
EX-396  
EX-397  
EX-398  
EX-399  
EX-400  
EX-401  
EX-402  
EX-403  
EX-404  
EX-405  
EX-406  
EX-407  
EX-408  
EX-409  
EX-410  
EX-411  
EX-412  
EX-413  
EX-414  
EX-415  
EX-416  
EX-417  
EX-418  
EX-419  
EX-420  
EX-421  
EX-422  
EX-423  
EX-424  
EX-425  
EX-426  
EX-427  
EX-428  
EX-429  
EX-430  
EX-431  
EX-432  
EX-433  
EX-434  
EX-435  
EX-436  
EX-437  
EX-438  
EX-439  
EX-440  
EX-441  
EX-442  
EX-443  
EX-444  
EX-445  
EX-446  
EX-447  
EX-448  
EX-449  
EX-450  
EX-451  
EX-452  
EX-453  
EX-454  
EX-455  
EX-456  
EX-457  
EX-458  
EX-459  
EX-460  
EX-461  
EX-462  
EX-463  
EX-464  
EX-465  
EX-466  
EX-467  
EX-468  
EX-469  
EX-470  
EX-471  
EX-472  
EX-473  
EX-474  
EX-475  
EX-476  
EX-477  
EX-478  
EX-479  
EX-480  
EX-481  
EX-482  
EX-483  
EX-484  
EX-485  
EX-486  
EX-487  
EX-488  
EX-489  
EX-490  
EX-491  
EX-492  
EX-493  
EX-494  
EX-495  
EX-496  
EX-497  
EX-498  
EX-499  
EX-500  
EX-501  
EX-502  
EX-503  
EX-504  
EX-505  
EX-506  
EX-507  
EX-508  
EX-509  
EX-510  
EX-511  
EX-512  
EX-513  
EX-514  
EX-515  
EX-516  
EX-517  
EX-518  
EX-519  
EX-520  
EX-521  
EX-522  
EX-523  
EX-524  
EX-525  
EX-526  
EX-527  
EX-528  
EX-529  
EX-530  
EX-531  
EX-532  
EX-533  
EX-534  
EX-535  
EX-536  
EX-537  
EX-538  
EX-539  
EX-540  
EX-541  
EX-542  
EX-543  
EX-544  
EX-545  
EX-546  
EX-547  
EX-548  
EX-549  
EX-550  
EX-551  
EX-552  
EX-553  
EX-554  
EX-555  
EX-556  
EX-557  
EX-558  
EX-559  
EX-560  
EX-561  
EX-562  
EX-563  
EX-564  
EX-565  
EX-566  
EX-567  
EX-568  
EX-569  
EX-570  
EX-571  
EX-572  
EX-573  
EX-574  
EX-575  
EX-576  
EX-577  
EX-578  
EX-579  
EX-580  
EX-581  
EX-582  
EX-583  
EX-584  
EX-585  
EX-586  
EX-587  
EX-588  
EX-589  
EX-590  
EX-591  
EX-592  
EX-593  
EX-594  
EX-595  
EX-596  
EX-597  
EX-598  
EX-599  
EX-600  
EX-601  
EX-602  
EX-603  
EX-604  
EX-605  
EX-606  
EX-607  
EX-608  
EX-609  
EX-610  
EX-611  
EX-612  
EX-613  
EX-614  
EX-615  
EX-616  
EX-617  
EX-618  
EX-619  
EX-620  
EX-621  
EX-622  
EX-623  
EX-624  
EX-625  
EX-626  
EX-627  
EX-628  
EX-629  
EX-630  
EX-631  
EX-632  
EX-633  
EX-634  
EX-635  
EX-636  
EX-637  
EX-638  
EX-639  
EX-640  
EX-641  
EX-642  
EX-643  
EX-644  
EX-645  
EX-646  
EX-647  
EX-648  
EX-649  
EX-650  
EX-651  
EX-652  
EX-653  
EX-654  
EX-655  
EX-656  
EX-657  
EX-658  
EX-659  
EX-660  
EX-661  
EX-662  
EX-663  
EX-664  
EX-665  
EX-666  
EX-667  
EX-668  
EX-669  
EX-670  
EX-671  
EX-672  
EX-673  
EX-674  
EX-675  
EX-676  
EX-677  
EX-678  
EX-679  
EX-680  
EX-681  
EX-682  
EX-683  
EX-684  
EX-685  
EX-686  
EX-687  
EX-688  
EX-689  
EX-690  
EX-691  
EX-692  
EX-693  
EX-694  
EX-695  
EX-696  
EX-697  
EX-698  
EX-699  
EX-700  
EX-701  
EX-702  
EX-703  
EX-704  
EX-705  
EX-706  
EX-707  
EX-708  
EX-709  
EX-710  
EX-711  
EX-712  
EX-713  
EX-714  
EX-715  
EX-716  
EX-717  
EX-718  
EX-719  
EX-720  
EX-721  
EX-722  
EX-723  
EX-724  
EX-725  
EX-726  
EX-727  
EX-728  
EX-729  
EX-730  
EX-731  
EX-732  
EX-733  
EX-734  
EX-735  
EX-736  
EX-737  
EX-738  
EX-739  
EX-740  
EX-741  
EX-742  
EX-743  
EX-744  
EX-745  
EX-746  
EX-747  
EX-748  
EX-749  
EX-750  
EX-751  
EX-752  
EX-753  
EX-754  
EX-755  
EX-756  
EX-757  
EX-758  
EX-759  
EX-760  
EX-761  
EX-762  
EX-763  
EX-764  
EX-765  
EX-766  
EX-767  
EX-768  
EX-769  
EX-770  
EX-771  
EX-772  
EX-773  
EX-774  
EX-775  
EX-776  
EX-777  
EX-778  
EX-779  
EX-780  
EX-781  
EX-782  
EX-783  
EX-784  
EX-785  
EX-786  
EX-787  
EX-788  
EX-789  
EX-790  
EX-791  
EX-792  
EX-793  
EX-794  
EX-795  
EX-796  
EX-797  
EX-798  
EX-799  
EX-800  
EX-801  
EX-802  
EX-803  
EX-804  
EX-805  
EX-806  
EX-807  
EX-808  
EX-809  
EX-810  
EX-811  
EX-812  
EX-813  
EX-814  
EX-815  
EX-816  
EX-817  
EX-818  
EX-819  
EX-820  
EX-821  
EX-822  
EX-823  
EX-824  
EX-825  
EX-826  
EX-827  
EX-828  
EX-829  
EX-830  
EX-831  
EX-832  
EX-833  
EX-834  
EX-835  
EX-836  
EX-837  
EX-838  
EX-839  
EX-840  
EX-841  
EX-842  
EX-843  
EX-844  
EX-845  
EX-846  
EX-847  
EX-848  
EX-849  
EX-850  
EX-851  
EX-852  
EX-853  
EX-854  
EX-855  
EX-856  
EX-857  
EX-858  
EX-859  
EX-860  
EX-861  
EX-862  
EX-863  
EX-864  
EX-865  
EX-866  
EX-867  
EX-868  
EX-869  
EX-870  
EX-871  
EX-872  
EX-873  
EX-874  
EX-875  
EX-876  
EX-877  
EX-878  
EX-879  
EX-880  
EX-881  
EX-882  
EX-883  
EX-884  
EX-885  
EX-886  
EX-887  
EX-888  
EX-889  
EX-890  
EX-891  
EX-892  
EX-893  
EX-894  
EX-895  
EX-896  
EX-897  
EX-898  
EX-899  
EX-900  
EX-901  
EX-902  
EX-903  
EX-904  
EX-905  
EX-906  
EX-907  
EX-908  
EX-909  
EX-910  
EX-911  
EX-912  
EX-913  
EX-914  
EX-915  
EX-916  
EX-917  
EX-918  
EX-919  
EX-920  
EX-921  
EX-922  
EX-923  
EX-924  
EX-925  
EX-926  
EX-927  
EX-928  
EX-929  
EX-930  
EX-931  
EX-932  
EX-933  
EX-934  
EX-935  
EX-936  
EX-937  
EX-938  
EX-939  
EX-940  
EX-941  
EX-942  
EX-943  
EX-944  
EX-945  
EX-946  
EX-947  
EX-948  
EX-949  
EX-950  
EX-951  
EX-952  
EX-953  
EX-954  
EX-955  
EX-956  
EX-957  
EX-958  
EX-959  
EX-960  
EX-961  
EX-962  
EX-963  
EX-964  
EX-965  
EX-966  
EX-967  
EX-968  
EX-969  
EX-970  
EX-971  
EX-972  
EX-973  
EX-974  
EX-975  
EX-976  
EX-977  
EX-978  
EX-979  
EX-980  
EX-981  
EX-982  
EX-983  
EX-984  
EX-985  
EX-986  
EX-987  
EX-988  
EX-989  
EX-990  
EX-991  
EX-992  
EX-993  
EX-994  
EX-995  
EX-996  
EX-997  
EX-998  
EX-999  
EX-1000

Docket No. RSW920010142US1

consistent method for finding and accessing this information, not necessarily for the user, but mostly for the user's Web "browser". A browser is a program capable of submitting a request for information identified by an 5 identifier, such as, for example, a URL. A user may enter a domain name through a graphical user interface (GUI) for the browser to access a source of content. The domain name is automatically converted to the Internet Protocol (IP) address by a domain name system (DNS), which is a service 10 that translates the symbolic name entered by the user into an IP address by looking up the domain name in a database.

The Internet also is widely used to transfer applications to users using browsers. With respect to commerce on the Web, individual consumers and business use 15 the Web to purchase various goods and services. In offering goods and services, some companies offer goods and services solely on the Web while others use the Web to extend their reach.

Content distribution systems are employed by 20 businesses and entities delivering content, such as Web pages or files to users on the Internet. Currently, content providers will set up elaborate server systems or other types of data sources to provide content to various users. Web content distribution systems are those 25 systems that are employed to distribute content to these servers and caches. This type of setup includes various nodes that act as sources of data. In this type of content distribution scheme, data from a primary or

Docket No. RSW920010142US1

publishing node is propagated to all of the other nodes in the system. These types of systems cache or hold content for distribution to requestors at clients, such as personal computers and personal digital assistants.

5 Different mechanisms are employed to determine whether the content cached at the node is current and whether this content should be distributed. Currently, content providers are required to use content distribution systems in which the same type of mechanism is used to  
10 determine whether the content is current. Additionally, if a content provider sends content to a non-content distribution capable system, the content is formatted in a manner differently than in those for content distribution capable systems.

15 Therefore, it would be advantageous to have an improved method, apparatus, and computer-implemented instructions for caching content in a node.

#### SUMMARY OF THE INVENTION

The present invention provides a method, apparatus,  
20 and computer implemented instructions for managing data in a network data processing system. A packet containing data associated with content is received. A determination is made as to whether the packet is enabled for content distribution by examining the data packet. Responsive to  
25 the packet being enabled for content distribution, the

Docket No. RSW920010142US1

content is distributed in response to a request for the content without requiring a validity check. If the packet is not enabled for content distribution, a validity check is performed on the content using control information contained within the header of the data packet.

#### BRIEF DESCRIPTION OF THE DRAWINGS

The novel features believed characteristic of the invention are set forth in the appended claims. The invention itself, however, as well as a preferred mode of use, further objectives and advantages thereof, will best be understood by reference to the following detailed description of an illustrative embodiment when read in conjunction with the accompanying drawings, wherein:

15       Figure 1 is a network data processing system in accordance with a preferred embodiment of the present invention;

Figure 2 is a block diagram of a data processing system that may be implemented as a server in accordance with a preferred embodiment of the present invention;

Figure 3 is a diagram illustrating data flow in updating content at data sources in accordance with a preferred embodiment of the present invention;

Docket No. RSW920010142US1

**Figure 4** is a diagram illustrating a data packet in accordance with a preferred embodiment of the present invention;

5 **Figure 5** is a flowchart of a process for receiving content from a content provider in accordance with a preferred embodiment of the present invention;

**Figure 6** is a flowchart of a process for receiving content in accordance with a preferred embodiment of the present invention; and

10 **Figure 7** is a flowchart of a process for handling a request for content at a node in accordance with a preferred embodiment of the present invention.

#### **DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT**

With reference now to the figures and in particular 15 to **Figure 1**, a network data processing system is depicted in accordance with a preferred embodiment of the present invention. Network data processing system 100 in this example includes network 102, which interconnects servers 104, 106, 108, 110, 124, and 126. These servers provide 20 content to clients, such as clients 112, 114, and 116, through network 102. In this example, network 102 takes the form of the Internet.

Servers 104-110 are servers within a Web content distribution system. This system also includes content

PRINTED IN U.S.A. 100-1000

Docket No. RSW920010142US1

management and creator 118, which is connected to server 110 by local area network (LAN) 120. This Web content distribution system is also referred to as a content distribution framework and is an example of a system in which inconsistency between data and data sources is minimized, such as servers 104-108. In this example, server 110 functions as a primary publishing node while servers 104-108 serve as data sources to provide content to users making requests. Server 110 includes a master content distribution server and a master content distribution (CD) server process 122.

Master content distribution server process 122 accepts notifications of new, deleted, or modified content from content management and creator 118. These notifications are propagated to servers 104-108, which then can invalidate or pull updated content from various sources. The content may be pulled from server 110 or from other sources.

Typically, when a content publisher issues a notification to master CD server 122 in server 110, an identification of a staging server containing the content is made. Each of the servers pulling content includes a content distribution process (not shown), which will update content on a server when a notification is received.

In these examples, the servers act as content distribution capable caches. CD-capable caches subscribe to content from specific providers that are equipped with the capability to issue notifications; this subscription mechanism could be enhanced with "content groups", where a

TOP SECRET//COMINT

Docket No. RSW920010142US1

certain set of content is tagged as belonging to a content group. These tags may be provided by the content creator, or inferred based on regular expression matching on the URL (e.g., SPORTS content group could be defined as all URLs

5 that match www.espn.com/mlb/\*, www.espn.com/nba/\*, www.espn.com/nfl/\*, www.espn.com/nhl/\*, and www.espn.com/sports/headlines/\*.html).

This framework may be used to distribute multiple content types. For example, the framework may be used to move static content. Additionally, the framework may be

10 used to publish or present documents on Web sites. In this instance, the framework will send notifications to the various nodes from the publishing node. The framework takes up the responsibility of updating the various repositories. Next, the framework may be used to

15 move applications to the nodes for distribution and use.

Third, the framework may be used to manage cached dynamic content. Finally, the framework may be used to distribute media files. Media files are similar to static pages. However, their large size requires a slightly

20 different treatment. The transport mechanism in the framework may include mechanisms to pace the data distribution depending on factors such as the media type, the bandwidth requirements, and available bandwidth.

Network data processing system 100 includes servers, 25 which may be either content distribution capable or content distribution incapable. For example, server 124 and server 126 are content incapable servers in these examples. In other words, notifications sent out to

100-0000000000000000

Docket No. RSW920010142US1

network 102 cannot be used by these servers to receive notifications that the content has been updated or to pull updated content in response to the notifications.

These providers should also expect that their data may  
5 be cached at both CD-capable and CD-incapable caches, such as those described above. One problem, from the Web server perspective, is to define a protocol such that correct behavior is seen at both kinds of caches, with minimal work by a content provider. At a CD-capable cache, content from  
10 CD-capable providers as well as content from CD-incapable providers co-exists. The challenge, from a caching perspective, is to devise cacheability criteria that works efficiently for content (from CD-capable providers) that this cache has subscribed to, and that works correctly for  
15 content that this cache has not subscribed to and for content from CD-incapable providers.

In solving the problem with caching content at both content capable and content incapable caches, the present invention provides a method, apparatus, and computer  
20 implemented instructions for caching or storing content in nodes in a network data processing system in a manner that works correctly for subscribed content in a cache, non-subscribed content in a cache, and for content distribution incapable providers. The mechanism of the  
25 present invention employs headers and cache control extensions to provide an ability to handle data at both content distribution capable and content distribution

RECEIVED - 2024 MAR 12 PM 10:00  
U.S. PATENT AND TRADEMARK OFFICE

Docket No. RSW920010142US1

incapable caches. In these examples, the headers are implemented as HTTP 1.1 headers.

When a CD-capable (provider) server sends back a response to a requestor (which could be an intermediary proxy cache or a browser), this server will add a new extension to the cache control header that says that the content that it is sending out is "CD-capable". If the intermediary is a CD-capable proxy cache, the intermediary will check if that specific page is being subscribed to at this node. If so, the intermediary will cache the page along with the extension header. If the intermediary does not subscribe to the page, it will delete the extension header and then cache the content.

When a subsequent request for the same page arrives at the cache, the cache will look at the cache-control headers and perform a validity check by determining if the factors indicate that the item is valid. These factors may be, for example, max-age, must-revalidate, proxy-revalidate, no-cache, or an Expires header. Since the cache is a CD-capable cache and the item is a CD-capable item, the cache can override these standard HTTP 1.1 cache-control headers and the Expires header and declare that the page is valid and send it out from the cache. The standard cache-control headers specified at the server ensure that the caching behavior at CD-incapable caches will be correct. But since CD-capable caches are equipped to receive notifications for subscribed data, they can choose to ignore the cache-

PENDING DRAFT

Docket No. RSW920010142US1

control headers and Expires header and pass the page on to the requestor.

Referring to **Figure 2**, a block diagram of a data processing system that may be implemented as a server, such as server 104 in **Figure 1**, is depicted in accordance with a preferred embodiment of the present invention. Data processing system 200 may be a symmetric multiprocessor (SMP) system including a plurality of processors 202 and 204 connected to system bus 206. Alternatively, a single processor system may be employed. Also connected to system bus 206 is memory controller/cache 208, which provides an interface to local memory 209. I/O bus bridge 210 is connected to system bus 206 and provides an interface to I/O bus 212. Memory controller/cache 208 and I/O bus bridge 210 may be integrated as depicted.

Peripheral component interconnect (PCI) bus bridge 214 connected to I/O bus 212 provides an interface to PCI local bus 216. A number of modems may be connected to PCI local bus 216. Typical PCI bus implementations will support four PCI expansion slots or add-in connectors. Communications links to clients 108-112 in **Figure 1** may be provided through modem 218 and network adapter 220 connected to PCI local bus 216 through add-in boards.

Additional PCI bus bridges 222 and 224 provide interfaces for additional PCI local buses 226 and 228, from which additional modems or network adapters may be supported. In this manner, data processing system 200 allows connections to multiple network computers. A

08562000000000000000000000000000

Docket No. RSW920010142US1

memory-mapped graphics adapter 230 and hard disk 232 may also be connected to I/O bus 212 as depicted, either directly or indirectly.

Those of ordinary skill in the art will appreciate 5 that the hardware depicted in **Figure 2** may vary. For example, other peripheral devices, such as optical disk drives and the like, also may be used in addition to or in place of the hardware depicted. The depicted example is not meant to imply architectural limitations with respect 10 to the present invention.

The data processing system depicted in **Figure 2** may be, for example, an IBM e-Server pSeries system, a product of International Business Machines Corporation in Armonk, New York, running the Advanced Interactive Executive (AIX) 15 operating system or LINUX operating system.

With reference now to **Figure 3**, a diagram illustrating data flow in updating content at data sources is depicted in accordance with a preferred embodiment of the present invention. In this example, content at Web 20 server 300 and Web server 302 is updated from content located at originating Web server 304. These servers are servers in a Web content distribution system such as that illustrated in **Figure 1**. Web server 300 includes temporary storage 306 and available content 308. 25 Similarly, Web server 302 includes temporary storage 310 and available content 312.

When a user requests content from a client, such as client 314, the request is typically made from a browser,

Docket No. RSW920010142US1

such as browser 316. The request may be routed to either Web server 300 or Web server 302 through a load balancing system. If Web server 300 receives the request, the content returned to client 314 is returned from content in available content 308. This content may be, for example, a Web page or an audio file. If the request is routed to Web server 302, the content is returned to client 314 from content in available content 312. In either case, the content is identical.

At some point, changes to the content in available content 308 and available content 312 may be made. For example, a new Web page may be added, a Web page may be modified, or a Web page may be deleted from the content. The initiation of this process occurs when a signal indicating that content is to be updated is received by Web server 300 and Web server 302. This signal is received from originating Web server 304 in this example. In these examples, Web server 300 and Web server 302 pull the content from originating Web server 304. The content is stored in temporary storage 306 and temporary storage 310 during the pull process. When Web server 300 receives all of the new content, this Web server sends an acknowledgment signal back to originating Web server 304. Similarly, Web server 302 will transmit an acknowledgment signal to originating Web server 304 when Web server 302 has pulled all of the new content. The completion of the pulling of new content may occur at different times in Web server 300 and Web server 302 depending on the various

Docket No. RSW920010142US1

network conditions, such as available bandwidth, network traffic, and the number of hops to originating Web server 304.

This content is not made available to clients until a 5 second signal is received from originating Web server 304 indicating that the content is to be published or made available in response to request from clients. During this time, the content in available content 308 and available content 312 is used to reply to requests from 10 clients.

In addition, Web server 300 and Web server 302 both validate content for distribution based on notifications from a server, such as originating Web server 304. In these examples, content received from originating Web 15 server 304 by Web server 300 for Web server 302 includes an indicator, such as an extension to the cache control header, to identify the content as being content distribution capable. These Web servers check the extension and the data packet carrying the content to see 20 whether the content is subscribed to at the servers. If the content is subscribed to, the content is saved at the servers along with the header information. Otherwise, the header is deleted and the content is cached. This header information, especially the indicator, is used by Web 25 server 300 and Web server 302 to determine whether the content may be served or distributed to a requestor without performing a more typical validity check. A typical validity check compares the current date and time

Docket No. RSW920010142US1

to the Expires header of the page to see if it is still valid. The Expires header indicates when a page expires or becomes invalid. In making the check, the server also examines other cache control directives, such as,

5 for example, must-revalidate, to see if it can serve out the page. The setting of a must-revalidate header requires the server or cache to contact the origin server to see if the cached content is still valid. A requesting client browser also may specify a desired max-age, max-  
10 stale, min-fresh times, and validity checks are performed against the cached content to see if the page adheres to the requirements of the client.

If the content is received by a server that is content distribution incapable, the indicator is ignored  
15 by the server. In this case, the server performs the normal validity checks.

Turning next to **Figure 4**, a diagram illustrating a data packet is depicted in accordance with a preferred embodiment of the present invention. Data packet **400** is  
20 an example of a data packet in which content control information has been included to identify the data within data packet **400** as being content distribution capable. Data packet **400** includes a header **402** and a payload **404**. Header **402** includes cache control information **406** and  
25 indicator **408**. In this example, indicator **408** identifies content **410** within payload **404** as being content distribution capable data.

Docket No. RSW920010142US1

Cache control headers are used to specify how cache content is to be handled. For example, cache control headers may be specified as follows: cache control: max-age=<blah>, no-transform, must-revalidate i.e., as a sequence of directives, which can stand by themselves (must-revalidate) or associated with a value(max-age).  
5 In these examples, two directives or cache control headers are added. These two directives are CDIST\_CDN = <value> and CDIST\_FILENAME = <value>. The presence of  
10 these directives tell the cache that the origin server is a content distribution capable server. The directives also carry information that is valuable for use in maintaining state about URLs and the file names where they are stored. These examples are merely illustrative  
15 and not limiting to the types of headers or directives that may be used to inform a cache about content distribution capability.

Cache control information 406 in header 402 is, in these examples, standard cache control information to 20 allow content distribution incapable caches to correctly handle content 410. Content distribution capable caches may choose to ignore most cache control information 406. Some cache control directives such as "no-store" have stringent semantics that prohibit a cache from ignoring  
25 them.

With reference now to **Figure 5**, a flowchart of a process for receiving content from a content provider, is depicted in accordance with a preferred embodiment of the

Docket No. RSW920010142US1

present invention. The process illustrated in **Figure 5** may be implemented in a content provider, such as originating Web server 304 in **Figure 3**.

The process begins by receiving a request from the 5 requestor (step 500). This request may be, for example, a request to pull content. An indicator is added to cache the control header of a data packet (step 502). This indicator may be, for example, indicator 408 in **Figure 4**. The content is placed into the data packet (step 504). 10 This content may be, for example, data for a Web page. The data packet is sent to the requestor (step 506). Next, a determination is made as to whether there is more content to be sent (step 508).

In step 508, if no more content is present, the 15 process terminates. With reference again to step 508, if a determination is made that there is more content, the process returns to step 502, as described above.

Turning next to **Figure 6**, a flowchart of a process for receiving content is depicted in accordance with a 20 preferred embodiment of the present invention. The process illustrated in **Figure 6** may be implemented in a Web server, such as Web server 300 in **Figure 3** from a content provider, such as originating Web server 304 in **Figure 3**.

The process begins by receiving a data packet (step 25 600). The data packet is parsed (step 602). Next, a determination is made as to whether the data is subscribed to by a node (step 604). If the data is subscribed to by

Docket No. RSW920010142US1

a node, the data is cached with the cache control header (step 606) and the process terminates thereafter.

Turning again to step 604, if the data is not subscribed to by a node, the header is deleted (step 608).

5 The data is cached (step 610) and the process terminates thereafter. With respect to data not subscribed to by a node, the following example provides a further explanation. Assume a company called foobar.com hosts both NFL and World Soccer news and scores. In this  
10 example, a cache is installed in Europe and subscribes to the SOCCER content group alone, containing URLs www.foobar.com/soccer/\*. Now, it is possible that someone in Europe requests a page "www.foobar.com/nfl/headlines.html". If that page is not  
15 present in the cache, the cache will request the page from the origin server, cache the page, and deliver the page to the client. Even though the cache does not subscribe to that page, the page is placed into the cache via a request/response.

20 With reference now to **Figure 7**, a flowchart of a process for handling a request for content at a node is depicted in accordance with a preferred embodiment of the present invention. The process illustrated in **Figure 7** may be implemented in a node, such as Web server 300 in  
25 **Figure 3**.

The process begins by receiving a request for content (step 700). This request is received from a user at a client, such as a personal computer or a personal digital

Docket No. RSW920010142US1

assistant. The cache control header associated with content is examined (step 702). The cache control header includes information from a header, such as header 402 in **Figure 4**. Then, a determination is made as to whether an 5 indicator is present (step 704). This indicator may be, for example, indicator 408 in **Figure 4**. If an indicator is present, the content is identified as valid (step 706). The content is sent to the requestor (step 708), with the process terminating thereafter.

10       Returning to step 704, if an indicator is not present, a validity check is performed (step 710). Next, a determination is made as to whether the content is valid (step 712). If the content is valid, the process returns to step 706, as described above. In step 712, if a 15 determination is made that the content is not valid, the process terminates.

      Thus, the present invention provides a method, apparatus, and computer implemented instructions for caching subscribed and non-subscribed content. Using the 20 mechanism of the present invention, a content distribution capable cache which subscribes to a subset of content served from content distribution capable servers can cache at a higher efficiency for content subscribed to by the cache. The main efficiencies achieved using the 25 mechanism of the present invention are due to the fact that the often incorrect Expires: header and the cache control directives are ignored. More often than not, Web administrators will not be able to specify when a document

Docket No. RSW920010142US1

"expires". Typically, administrators are either conservative, setting a short expiration time, causing caches to not serve out perfectly valid content from their repository; or they are aggressive, setting a long

5 expiration time, causing the caches to serve out stale content. The mechanism of the present invention allows caches to selectively ignore Expires headers and cache control directives, thus enhancing the number of pages that a cache can directly serve out to clients instead of  
10 having to proxy back to an origin server. Clients then see a better "hit rate", and a reduction in the average latency seen in responses from the cache. Additionally, the cache also may cache other content, thus functioning as a regular Web intermediary for such content. However,  
15 for non-subscribed or content distribution incapable content, the cache strictly enforces the cache-control headers.

Using the mechanism of the present invention, a content distribution-incapable cache will work just as  
20 before, following the semantics laid down by the cache-control headers. Further, the mechanism of the present invention minimizes the work required from an administrator of a Web server. With the mechanism of the present invention, the administrator is only required to  
25 add a new cache-control extension, indicating that the content is content distribution capable, to the configuration, so that the server tacks that on to all the responses. In this manner, the administrator may be

F014750  
F014751  
F014752  
F014753  
F014754  
F014755

Docket No. RSW920010142US1

assured that the caching will work correctly across all kinds of intermediaries. As added functionality, the administrator may partition the content into content distribution capable content and add that header only to 5 those pages. This is a likely scenario because the administrator may not have the ability to issue update notifications for all types of content that the administrator may host.

The mechanism of the present invention also may be 10 used in architectures in which intermediate nodes are chained, and each node is either content distribution capable or content distribution incapable. This mechanism works with this type of architecture because all caches pass the headers along to the requestor in the chain.

15 Further, using the mechanism of the present invention, a cache will not ignore all cache-control extensions. For example, the cache may ignore time-based extensions, but may honor "no-cache" and "no-store". The information ignored or used depends on the particular 20 implementation.

It is important to note that while the present invention has been described in the context of a fully functioning data processing system, those of ordinary skill in the art will appreciate that the processes of 25 the present invention are capable of being distributed in the form of a computer readable medium of instructions and a variety of forms and that the present invention applies equally regardless of the particular type of

Docket No. RSW920010142US1

signal bearing media actually used to carry out the distribution. Examples of computer readable media include recordable-type media such a floppy disc, a hard disk drive, a RAM, CD-ROMs, and transmission-type media such as digital and analog communications links.

The description of the present invention has been presented for purposes of illustration and description, and is not intended to be exhaustive or limited to the invention in the form disclosed. Many modifications and variations will be apparent to those of ordinary skill in the art. For example, the illustrated embodiments are described with respect to a pull system in which nodes pull content from a source. The mechanism of the present invention also may be used with a push system in which content is pushed from a source to the nodes. The embodiment was chosen and described in order to best explain the principles of the invention, the practical application, and to enable others of ordinary skill in the art to understand the invention for various embodiments with various modifications as are suited to the particular use contemplated.

RECORDED - 09/21/2012